PsRBExportDevices PRO
Version 14.04

Pragnaan ReportBuilder Export Devices

User's Manual

The User's Manual is Copyright ©, 2000-2011 Pragnaan Software Group
Rarefind Engg. Innovations Pvt.Ltd.,
All Rights Reserved.

Table of Contents

Introduction....3
Installation of Trial Edition....3
Installation of Registered Edition....4
Re-Compiling/Re-Installing Registered Edition (using batch file)....4
Re-Compiling/Re-Installing Registered Edition (Manually)....12
Integrating Help Files into the IDE....18
Un-Installation....20
Architectural overview - Class Hierarchy....21
Localization....25
How to use Pragnaan ReportBuilder Export Devices?....26
Exporting Reports Programmatically....26
Exporting Reports to User Stream....27
EMailing Reports....27
Using PsRBExportDevices in Web Server Applications....28
WPRichText Support....30
Custom Report Components and PsRBExportDevices....30
Properties and Events explained....32
Ordering and Delivery....69
Contact....69

Introduction

The perfect ReportBuilder exporting companion! PsRBExportDevices is a suite of Export Devices for ReportBuilder. You can save crisp looking copies of your ReportBuilder reports as **PDF, HTML, XHTML, RTF, Excel, JPEG, GIF, BMP, EMF** and **WMF**. It is loaded with exciting new features such as run-time configuration dialogs, report E-Mailing support, watermarking, exporting to memory... the list goes on. It makes report exporting a real joy! PsRBExportDevices requires NO additional software. It is 100% native Object Pascal and compiles compactly into your executable file.

Installation of Trial Edition

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Uninstall any older versions of PsRBExportDevices, present on your computer.

Please visit http://www.rarefind.com/rbpro/downloads.html to download the trial version.

Make sure you have the appropriate executable file for your version of Delphi. After you have downloaded the executable file,

- Invoke the Setup program. The Setup program copies all files to a folder of your choice and automatically registers PsRBExportDevices into the development environment of **Delphi**.

Installation of Registered Edition

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Uninstall any older versions of PsRBExportDevices, present on your computer.

After you have downloaded the executable file,

- Invoke the Setup program. The Setup program copies all files to a folder of your choice and automatically registers PsRBExportDevices into the development environment of Delphi.

Note:

- Setup compiles the packages while installing. If the components fail to install for any reason, please refer to section "Re-compiling/Re-Installing Registered Edition".

Installing PsRBExportDevices for ReportBuilder version below 5.5

PsRBExportDevices supports ReportBuilder version 4.23 and above. By Default PsRBExportDevices is configured to work with ReportBuilder Version 5.5 and above. If you are using RB version below 5.5, follow the instructions below to install PsRBExportDevices.

1. Close Delphi
2. Open PsRBDefines.inc from <PsRBExportDevices>\Source folder. Add dot(.) before $DEFINE RB55Up to configure PsRBExportDevices to support ReportBuilder versions below 5.5. [i.e Change {$DEFINE RB55Up} to {.$DEFINE RB55Up}].
3. Save and Close PsRBDefines.inc
4. Re-compile PsRBExportDevices using Batch Files.

Re-Compiling/Re-Installing Registered Edition (using batch file)

In most cases, the installation program will completely install PsRBExportDevices without any intervention. If, however, the PsRBExportDevices components do not show up on the component palette, follow the steps below to re-install the package.

You may follow the same steps below to re-compile the PsRBExportDevices package if necessary.

Delphi 7

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the `<PsRBExportDevices>\Lib\D7` path under Library Path of Environment Options. Default path is `C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D7`
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file '`PsRBDefines.inc`' under `<PsRBExportDevices>\Source` to enable them.

Compiling:

- Close Delphi.
- Open the `<PsRBExportDevices>\Source` folder and run the file '`COMPILED7.bat`'.
- Run Delphi.

If the 'Ps RBuilder Export' tab is not present on the component palette, follow the installation instructions below.

Installing:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with `<PsRBExportDevices>\Lib\D7`
- Choose Tools|Environment Options... on the main menu.
- Select the Library tab.
- Check under Library Path under Directories.
  Example:
  If the original Library Path was "`$(DELPHI)\Lib`" and you installed PsRBExportDevices to
  "`C:\Program Files\Pragnaan\PsRBExportDevices 14.04`"
  you should see the new Library Path as
  "`$(DELPHI)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D7`"
- Choose Component|Install Packages...
- Click Add... and Open `DCLPsRBD70.bpl` from `<PsRBExportDevices>\Lib\D7`
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Delphi 2006

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the `<PsRBExportDevices>\Lib\D10` path under Library Path of Environment Options. Default path is `C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D10`
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file '`PsRBDefines.inc`' under `<PsRBExportDevices>\Source` to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D10
- Choose Tools|Options... on the main menu.
- Select the Delphi Options|Library – Win32.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(BDS)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices `14.04`"
  you should see the new Library Path as
  "$(BDS)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices `14.04`\Lib\D10"
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD10.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD10.bpl from <PsRBExportDevices>\Lib\D10 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD10.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD10.bpl from <PsRBExportDevices>\Lib\D10
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Delphi 2007

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the <PsRBExportDevices>\Lib\D11 path under Library Path of Environment Options. Default path is C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D11
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file 'PsRBDefines.inc' under <PsRBExportDevices>\Source to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D11
- Choose Tools|Options... on the main menu.
- Select the Delphi Options|Library – Win32.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(BDS)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices 14.04"
  you should see the new Library Path as
  "$(BDS)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D11"
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD11.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD11.bpl from <PsRBExportDevices>\Lib\D11 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD11.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD11.bpl from <PsRBExportDevices>\Lib\D11
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Delphi 2009

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the <PsRBExportDevices>\Lib\D12 path under Library Path of Environment Options. Default path is C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D12
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file 'PsRBDefines.inc' under <PsRBExportDevices>\Source to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D12
- Choose Tools|Options... on the main menu.
- Select the Delphi Options|Library – Win32.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(BDS)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices 14.04"
  you should see the new Library Path as
  "$(BDS)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D12"
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD12.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD12.bpl from <PsRBExportDevices>\Lib\D12 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD12.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD12.bpl from <PsRBExportDevices>\Lib\D12
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Delphi 2010

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the <PsRBExportDevices>\Lib\D14 path under Library Path of Environment Options. Default path is C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D14
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file 'PsRBDefines.inc' under <PsRBExportDevices>\Source to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D14
- Choose Tools|Options... on the main menu.
- Select the Delphi Options|Library – Win32.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(BDS)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices 14.04"
  you should see the new Library Path as
  "$(BDS)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D14"
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD14.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD14.bpl from <PsRBExportDevices>\Lib\D14 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD14.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD14.bpl from <PsRBExportDevices>\Lib\D14
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Delphi XE

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the <PsRBExportDevices>\Lib\D15 path under Library Path of Delphi Environment Options. Default path is C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D15
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file 'PsRBDefines.inc' under <PsRBExportDevices>\Source to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D15
- Choose Tools|Options... on the main menu.
- Select the Environment Options | Delphi Options | Library.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(BDS)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices 14.04"
  you should see the new Library Path as
  "$(BDS)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D15"
- Click OK to close
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD15.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD15.bpl from <PsRBExportDevices>\Lib\D15 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD15.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD15.bpl from <PsRBExportDevices>\Lib\D15
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Delphi XE2

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the <PsRBExportDevices>\Lib\D16 path under Library Path of Delphi Environment Options. Default path is C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D16
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file 'PsRBDefines.inc' under <PsRBExportDevices>\Source to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D16
- Choose Tools|Options... on the main menu.
- Select the Environment Options | Delphi Options | Library.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(BDS)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices 14.04"
  you should see the new Library Path as
  "$(BDS)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D16"
- Click OK to close
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD16.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD16.bpl from <PsRBExportDevices>\Lib\D16 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD16.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD16.bpl from <PsRBExportDevices>\Lib\D16
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Re-Compiling/Re-Installing Registered Edition (manually)

In most cases, the installation program will completely install PsRBExportDevices without any intervention. If, however, the PsRBExportDevices components do not show up on the component palette, follow the steps below to re-install the package.

You may follow the same steps below to re-compile the PsRBExportDevices package if necessary.

Delphi 7

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the `<PsRBExportDevices>\Lib\D7` path under Library Path of Environment Options. Default path is `C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D7`
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file '`PsRBDefines.inc`' under `<PsRBExportDevices>\Source` to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D7
- Choose Tools|Environment Options... on the main menu.
- Select the Library tab.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(Delphi)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices `14.04`"
  you should see the new Library Path as
  "$(Delphi)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices `14.04`\Lib\D7"
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD70.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD70.bpl from <PsRBExportDevices>\Lib\D7 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD70.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD70.bpl from <PsRBExportDevices>\Lib\D7
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Delphi 2006

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the `<PsRBExportDevices>\Lib\D10` path under Library Path of Environment Options. Default path is `C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D10`
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file '`PsRBDefines.inc`' under `<PsRBExportDevices>\Source` to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D10
- Choose Tools|Options... on the main menu.
- Select the Delphi Options|Library – Win32.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(BDS)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices `14.04`"
  you should see the new Library Path as
  "$(BDS)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices `14.04`\Lib\D10"
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD10.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD10.bpl from <PsRBExportDevices>\Lib\D10 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD10.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD10.bpl from <PsRBExportDevices>\Lib\D10
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Delphi 2007

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the `<PsRBExportDevices>\Lib\D11` path under Library Path of Environment Options. Default path is `C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D11`
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file '`PsRBDefines.inc`' under `<PsRBExportDevices>\Source` to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D11
- Choose Tools|Options... on the main menu.
- Select the Delphi Options|Library – Win32.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(BDS)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices `14.04`"
  you should see the new Library Path as
  "$(BDS)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices `14.04`\Lib\D11"
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD11.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD11.bpl from <PsRBExportDevices>\Lib\D11 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD11.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD11.bpl from <PsRBExportDevices>\Lib\D11
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Delphi 2009

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the <PsRBExportDevices>\Lib\D12 path under Library Path of Environment Options. Default path is C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D12
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file 'PsRBDefines.inc' under <PsRBExportDevices>\Source to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D12
- Choose Tools|Options... on the main menu.
- Select the Delphi Options|Library – Win32.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(BDS)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices 14.04"
  you should see the new Library Path as
  "$(BDS)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D12"
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD12.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD12.bpl from <PsRBExportDevices>\Lib\D12 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD12.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD12.bpl from <PsRBExportDevices>\Lib\D12
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Delphi 2010

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the `<PsRBExportDevices>\Lib\D14` path under Library Path of Environment Options. Default path is `C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D14`
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file '`PsRBDefines.inc`' under `<PsRBExportDevices>\Source` to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D14
- Choose Tools|Options... on the main menu.
- Select the Delphi Options|Library – Win32.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(BDS)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices `14.04`"
  you should see the new Library Path as
  "$(BDS)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices `14.04`\Lib\D14"
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD14.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD14.bpl from <PsRBExportDevices>\Lib\D14 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD14.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD14.bpl from <PsRBExportDevices>\Lib\D14
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Delphi XE

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the <PsRBExportDevices>\Lib\D15 path under Library Path of Environment Options. Default path is C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D15
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file 'PsRBDefines.inc' under <PsRBExportDevices>\Source to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D15
- Choose Tools|Options... on the main menu.
- Select the Environment Options | Delphi Options | Library.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(BDS)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices 14.04"
  you should see the new Library Path as
  "$(BDS)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D15"
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD15.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD15.bpl from <PsRBExportDevices>\Lib\D15 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD15.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD15.bpl from <PsRBExportDevices>\Lib\D15
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Delphi XE2

IMPORTANT - before you proceed with this installation:

- Make sure you have the latest version of ReportBuilder installed (www.digital-metaphors.com).
- Make sure the Setup program has copied all files to a folder of your choice and included the <PsRBExportDevices>\Lib\D16 path under Library Path of Environment Options. Default path is C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D16
- PsRBExportDevices provides many useful additional features that needs to be enabled explicitly. Please open file 'PsRBDefines.inc' under <PsRBExportDevices>\Source to enable them.

Compiling:

- Run Delphi.
- Choose File|Close All on the main menu.
- Make sure the Library path is updated with <PsRBExportDevices>\Lib\D16
- Choose Tools|Options... on the main menu.
- Select the Environment Options | Delphi Options | Library.
- Check under Library Path under Directories..
  Example:
  If the original Library Path was "$(BDS)\Lib" and you installed PsRBExportDevices to "C:\Program Files\Pragnaan\PsRBExportDevices 14.04"
  you should see the new Library Path as
  "$(BDS)\Lib;C:\Program Files\Pragnaan\PsRBExportDevices 14.04\Lib\D16"
- Choose File|Open... on the main menu.
- Open the <PsRBExportDevices>\Source folder
- Select PsRBD16.dpk and click Open
- Click Compile on the package editor
- Copy PsRBD16.bpl from <PsRBExportDevices>\Lib\D16 to Windows\System32 (or WinNT\System32 if you are on WindowsNT/2000)
- Choose File|Open... on the main menu.
- Select DCLPsRBD16.dpk and click Open
- Click Compile on the package editor
- Choose File|Close All on the main menu.

Installing:

- Choose Component|Install Packages...
- Click Add... and Open DCLPsRBD16.bpl from <PsRBExportDevices>\Lib\D16
- Click OK.

A new tab names 'Ps RBuilder Export' should now appear on the component palette

Integrating Help Files into the IDE

Note: Windows Vista onwards, Microsoft has stopped bundling WinHelp with the OS. However, Microsoft has released a download of WinHlp32.exe for Windows Vista and Windows Server 2008 at http://support.microsoft.com/kb/917607

PsRBExportDevices component suite is shipped along with corresponding Help files to guide developers at every step. You can configure these help files to work seamlessly with your Delphi IDE. By doing this, you can get context sensitive help for any PsRBExportDevices' property, method or event, just by placing the cursor on the keyword and pressing F1. Please follow the step below to configure the Help System:

Delphi 6 and 7

1. Make sure the setup program has copied `PsRBExportDevices.hlp` and `PsRBExportDevices.cnt` files into your Delphi `..\Help` directory.
2. Start Delphi.
3. Select Help|Customize option (OpenHelp utility)
4. Select the "Index" tab.
5. Select Edit|Add Files.
6. Add `PsRBExportDevices.hlp` from Delphi `..\Help` directory.
7. Close OpenHelp utility.
8. When prompted – "Save changes to project?", click "Yes".
9. Delete any existing GID file in the Delphi Help directory (`..\Help\delphi5.GID` or `..\Help\delphi6.GID or Help\d7.GID`).

Un-Installation

1. Go to Start|Settings|Control Panel
2. Open 'Add/Remove Programs'
3. Remove 'Pragnaan ReportBuilder Export Devices' from the list
4. Un-install will complete

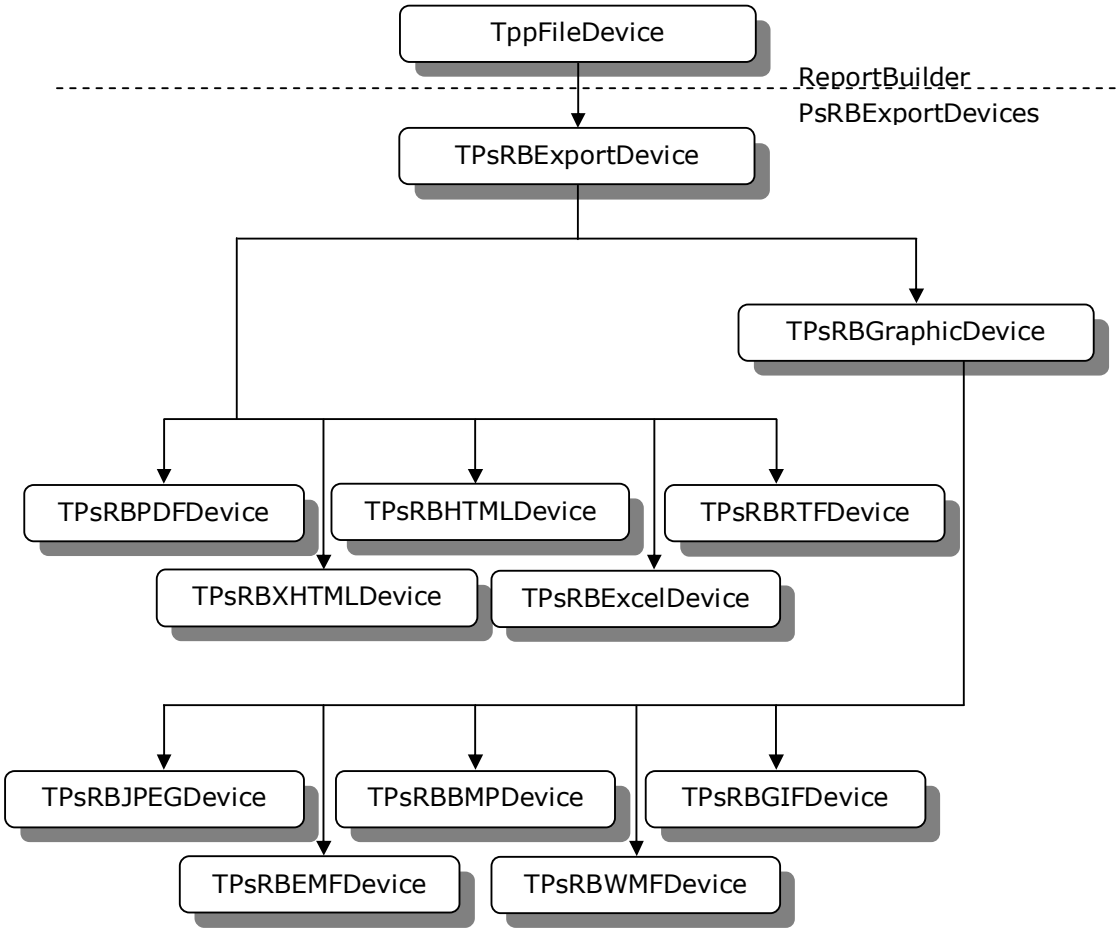
Note:

- To remove Library path and Browsing path information
  a. Open Delphi
  b. Choose Tools|Environment Options... on the main menu
  c. Select the Library tab
  d. Remove PsRBExportDevices path information from Library Path
  e. Remove PsRBExportDevices path information from Browsing Path
- The Un-install program may not remove newly created files or folders. Check and manually delete the folder where you installed PsRBExportDevices if required.

Architectural overview - Class Hierarchy

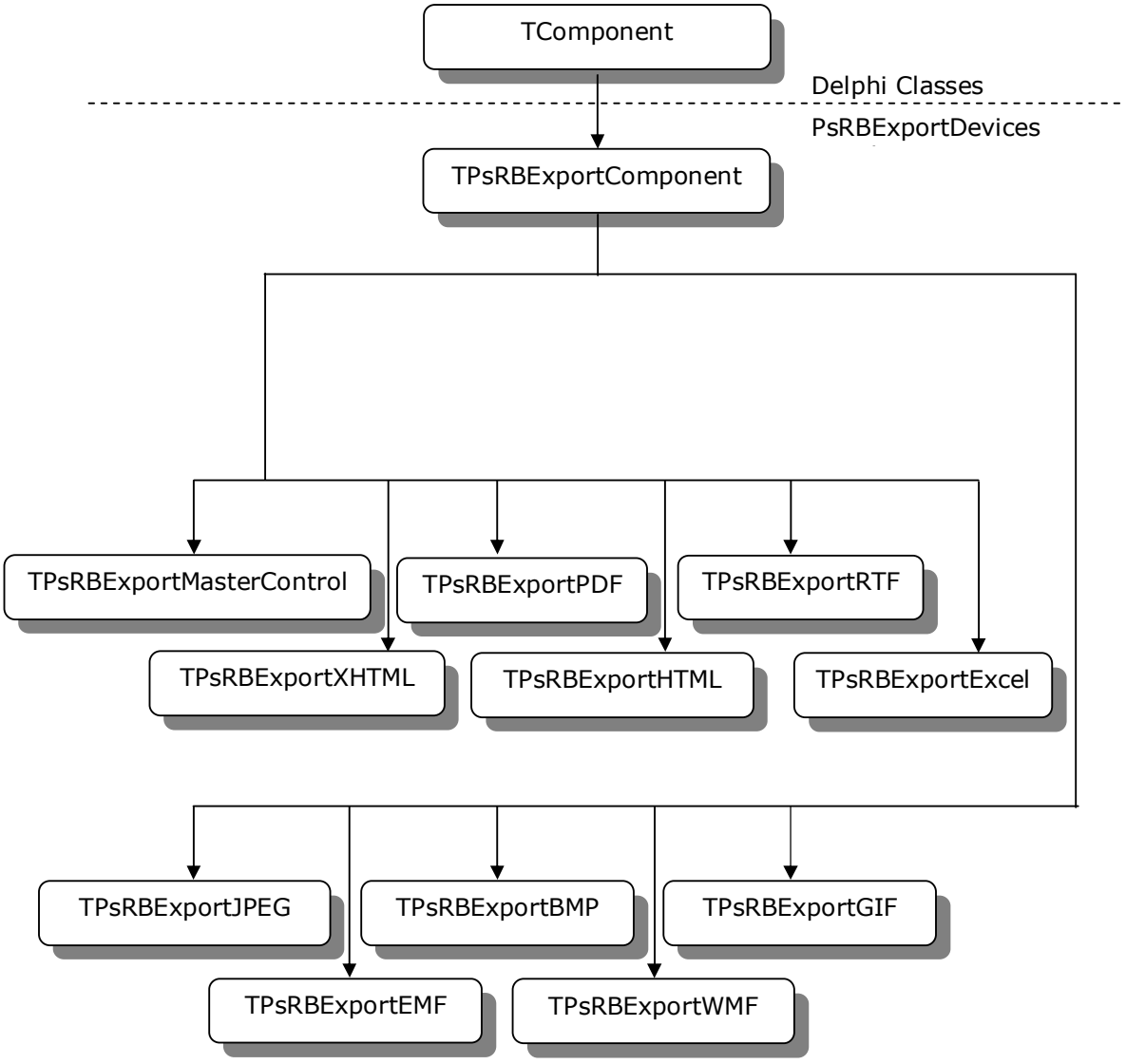
Device classes:

TPsRBExportDevice, derived from **TppFileDevice**, is the base class for all PsRBExportDevices. All instances of PsRBExportDevice classes in an application access and get their values from the respective global Settings object variables.



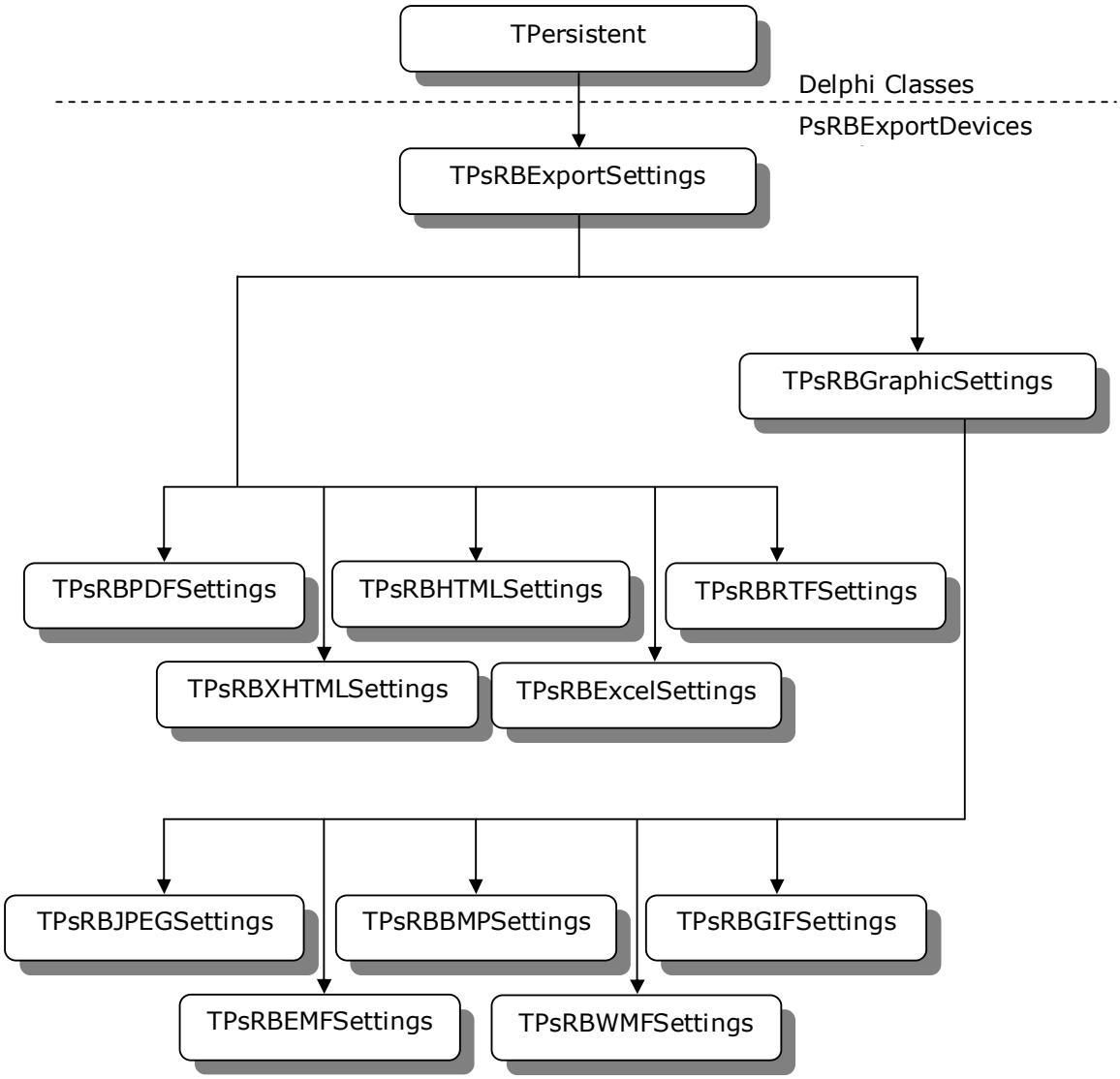
Component Classes:

TPsRBExportComponent, derived from **TComponent**, is the base class for all PsRBExportDevice Components. All instances of PsRBExport Components in an application access and get their values from the respective global Settings object variables.



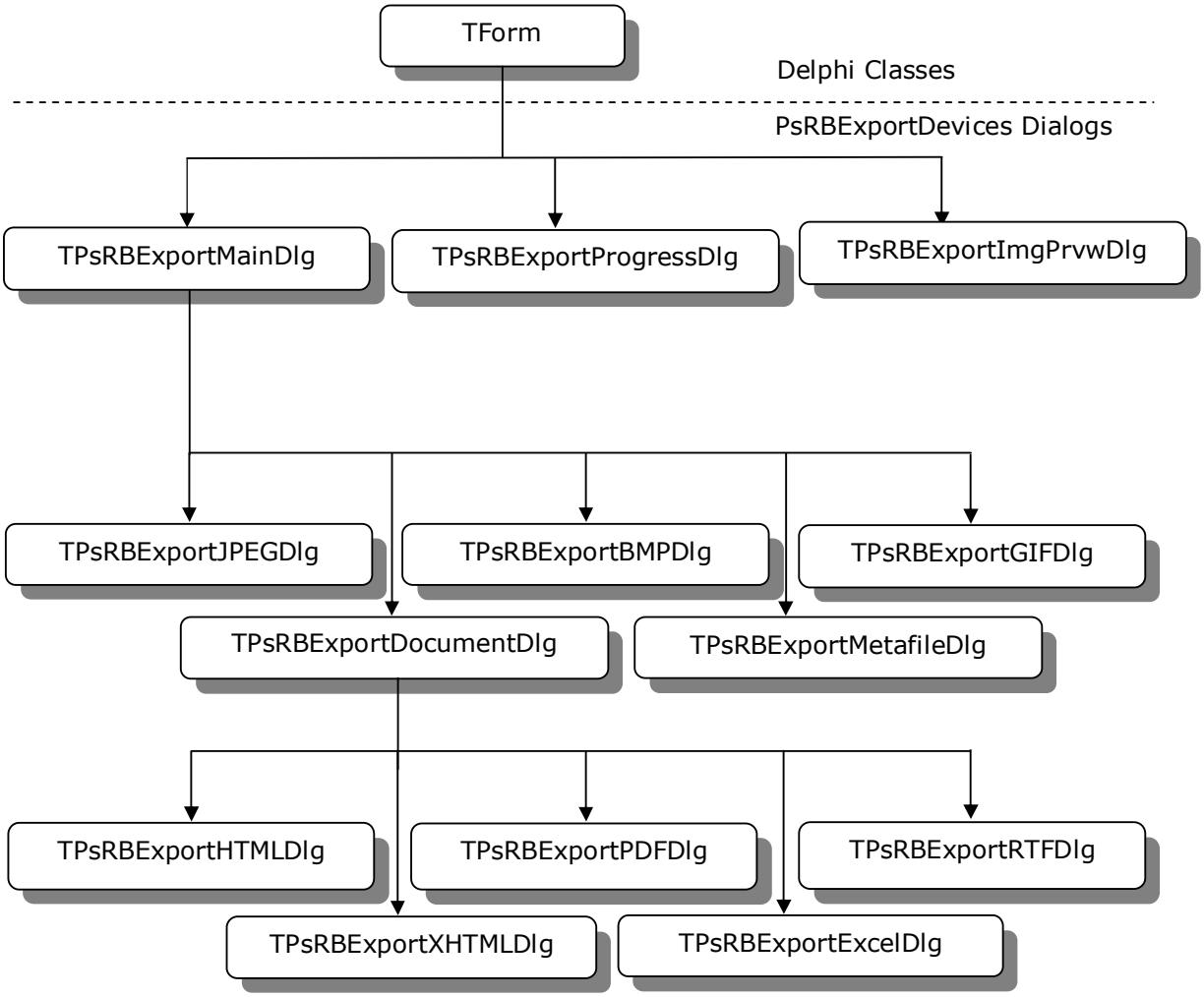
SettingsClass:

TPsRBExportSettings, derived from **TPersistent**, is the base class for all PsRBExportDevice Settings. All instances of PsRBExport Components and PsRBExportDevice classes in an application access and get their values from the respective global Settings object variables. As a result they all reflect the same values. Maintaining only one set of values also makes sure there is only one entry of a particular Device in the ReportBuilder and the save dialog.



Dialogs:

Apart from the above classes there are several more that are derived from TForm. These dialog classes manage the run-time configuration of PsRBExportDevices.



Localization

To simplify localization of the user-interface of the export devices, `PsRBExportDevices` uses resourcestrings throughout, wherever strings need to be displayed, including captions of controls on the dialogs and property default values. All these resourcestrings are located in the units `PsRBExport_Consts.pas` and `PsRBExport_DlgConsts`, if you need to change them. Resource files will be provided in the future.

How to use Pragnaan ReportBuilder Export Devices?

Drop dead simple! After successfully installation, a new tab "Ps RBuilder Export" will appear on the component palette. This tab will have several components: PsRBExportPDF, PsRBExportHTML, PsRBExportXHTML, PsRBExportRTF, PsRBExportExcel, PsRBExportJPEG, PsRBExportGIF, PsRBExportBMP, PsRBExportEMF and PsRBExportWMF. You can drop any or all of these components to achieve the corresponding export capability in your application.

Example: If you need PDF export capability in your application, just drop the PDF export component (PsRBExportPDF) from the component palette to your application. Your application will be able to save reports to PDF.

You can also drop the all-in-one PsRBExportMasterControl component which has all the supported file formats.

Exporting Reports Programmatically

Use the code below when you need to export reports programmatically. You would do this when you want the export to take place on an event automatically, without the user having to save the report through the preview window.

Delphi:

- Add the unit **PsRBRoutines** to the uses clause of the unit which need to Export
- Use the code below (The example exports to all supported formats. You may choose only the one you need):

```
procedure TForm1.SaveBtnClick(Sender: TObject);
begin
ExportToPDF(ppReport1, 'Sample.pdf');
ExportToHTML(ppReport1, 'Sample.htm');
ExportToXHTML(ppReport1, 'Sample.xht');
ExportToRTF(ppReport1, 'Sample.rtf');
ExportToExcel(ppReport1, 'Sample.xls');
ExportToGIF(ppReport1, 'Sample.gif');
ExportToJPEG(ppReport1, 'Sample.jpg');
ExportToBMP(ppReport1, 'Sample.bmp');
ExportToEMF(ppReport1, 'Sample.emf');
ExportToWMF(ppReport1, 'Sample.wmf');
end;
```

Exporting Reports to User Stream

Use the code below when you need to export reports programmatically and get the output in a TMemoryStream object.

Delphi:

- Add the unit **PsRBRoutines** to the uses clause of the unit which need to export
- Use the code below (The example exports a particular report to PDF):

```
procedure TForm1.PDFInStreamBtnClick(Sender: TObject);
var
MyStream: TMemoryStream;
begin
MyStream := TMemoryStream.Create;
ExportToPDFInStream(ppReport1, MyStream);
end;
```

EMailing Reports

Enabling EMail feature - Registered Users:
Follow the steps below to enable the EMailing feature in PsRBExportDevices.

- Enable the **EMailWithFastNet** or **EMailWithIndy** option in "PsRBDefines.inc" to get built-in support for FastNet or Indy Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
- Re-Compile PsRBExportDevices manually.

EMailing Reports:
Follow the steps below to EMail reports using PsRBExportDevices.

- Set **EMailAfterGenerate** to True.
- Provide a handler for **OnEMail** event to set the E-Mail information which will then be used to mail the report files. E-Mail is sent using either FastNet or Indy components.

Using PsRBExportDevices in Web Server Applications

Use the code below when you need Export Reports from a CGI Server Application.

- Create a new CGI application.
- Create a new form and name it **ReportForm**.
- Drop a **TppReport** component. Drop a **TPsRBExportPDF** component.
- Add an Action through the WebModule's Action Editor.
- Select the new action and create an OnAction event handler. Insert the code below for the OnAction event handler.
- Double-click on TppReport component and design the report as you do in any other application. Change any PsRBExportDevice properties if required.
- Choose Project|Options|Forms from the menu. Remove ReportForm from the Auto-create forms list.
- Add the unit of the ReportForm to the uses clause in the WebModule unit.
- Add the unit PsRBRoutines to the uses clause of the WebModule unit.
- Compile the application.
- Run it through your web browser.

Delphi:

- Add the unit **PsRBRoutines** to the uses clause of the WebModule unit

```
procedure TWebModule1.WebModule1WebActionItem1Action(Sender: TObject;
  Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
var
S: TMemoryStream;
begin
S := TMemoryStream.Create;
// REMOVE 'ReportForm' from the Auto-create forms list
with TReportForm.Create(nil) do
try
ExportToPDFInStream(ppReport1, S);
finally
Free;
end;
Response.ContentType := 'application/pdf';
Response.ContentStream := S;
// Do NOT free the stream,
// the response object will handle that task.
end;
```

Note: Commonly used MIME Media Types are listed below.

File Format	ContentType
PDF	application/pdf
HTML	text/html
RTF	application/rtf
EXCEL	application/excel
GIF	image/gif
JPEG	image/jpeg
BMP	image/bmp

Note: When exporting to HTML from a Web Server application, follow these tips:

1. Set SeparateFilePerPage property of PsRBExportHTML component to **False**.
2. Images, styles and script files will be created on the server. So, specify separate folders for these files. (Otherwise, all the files will be created in the cgi-bin directory.) Example:
   PsRBExportHTML.**ImageFolder** := '**images**';
   PsRBExportHTML.**CSSFileName** := '**styles\myreport.css**';
   PsRBExportHTML.**JavaScriptFileName** := '**scripts\myreport.js**';

   Make sure these folder have **READ** permission.

3. The images, styles and script files that are created on the server will **not** be deleted by PsRBExportDevices. Your application will have to periodically delete the files.

WPRichText Support

PsRBExportDevices now provides built in support for WPRichText. This feature is available only in the registered version of PsRBExportDevics. In order to use this feature the ReportBuilder support units for WPTools should be installed. Follow the steps below to enable WPRichText support:

1. Close Delphi
2. Open <PsRBExportDevices>\Source\PsRBDefines.inc file.
3. Enable the WPRichText option by removing the dot (.) prefixed to {.$DEFINE WPRichText}
4. Save and close PsRBDefines.inc
5. Re-Compile PsRBExportDevices Manually. Refer to section Re-Compiling/Re-Installing Manually - Registered Edition

Custom Report Components and PsRBExportDevices

Generally, any custom report component that you write can be made to appear in the exported document. Here's what happens: When PsRBExportDevices encounters a custom report component (a new TDrawCommand descendant) it provides an opportunity for the custom control to draw itself. It does this by calling the components Draw method. In the Draw method the control just needs to check for a few conditions and call appropriate PsRBExportDevices methods. The following examples show exactly what needs to be done.

Example 1: A rotated text control.

```
function TDrawRotatedText.Draw(aDevice: TppDevice): Boolean;
begin
…
if (aDevice is TppPrinterDevice) then
…
else if (aDevice is TppPrinterDevice) then
…
else if aDevice is TPsRBExportDevice then
with aDevice as TPsRBExportDevice do
begin
DrawRotatedText(Text, FontRotation, Left, Top, Width, Height, taAlignLeft,
  Font);
Result := True;
end
…
end;
```

Example 2: A rectangle with diagonals, exported as an image.

```
type
THackPsRBDevice = class(TPsRBExportDevice);

function TDrawInclined.Draw(aDevice: TppDevice): Boolean;
var
ADrawImage: TppDrawImage;
begin
…
if (aDevice is TppPrinterDevice) then
…
else if (aDevice is TppPrinterDevice) then
…
else if aDevice is TPsRBExportDevice then
begin
ADrawImage := TppDrawImage.Create(nil);
try
ADrawImage.Left := Left;
ADrawImage.Top := Top;
ADrawImage.Width := Width;
ADrawImage.Height := Height;
ADrawImage.Picture.Bitmap := TBitmap.Create;
ADrawImage.Picture.Bitmap.Width :=
  Round(THackPsRBDevice(aDevice).ThousandthOfMMToPixel(Width));
ADrawImage.Picture.Bitmap.Height :=
  Round(THackPsRBDevice(aDevice).ThousandthOfMMToPixel(Height));
ADrawImage.Picture.Bitmap.Canvas.Pen.Color := clBlack;
with ADrawImage.Picture.Bitmap do
begin
Canvas.MoveTo(0, 0);
Canvas.LineTo(0, Height - 1);
Canvas.LineTo(Width - 1, Height - 1);
Canvas.LineTo(Width - 1, 0);
Canvas.LineTo(0, 0);
Canvas.LineTo(Width - 1, Height - 1);
Canvas.MoveTo(0, Height - 1);
Canvas.LineTo(Width - 1, 0);
end;
THackPsRBDevice(aDevice).DrawImage(ADrawImage);
Result := True;
finally
ADrawImage.Free;
end
end
…
end;
```

Properties and Events explained

TPsRBExportComponent = class(TComponent)	
Properties - Run-time only	
ReportFileCount	**Type**: Integer ReportFileCount stores the total number of content files that were generated during the last export operation.
ReportFileNames	**Type**: TStringList ReportFileNames is a TStringList object that stores the names of all report files that were created during the last export operation. Read ReportFileNames if you need the file names of all report that were created.
OutputToUserStream	**Type**: Boolean **Default:** Boolean Specifies whether report should be output to user specified stream or to a file stream. Must be set to True, if you need to output the report into a user specified stream. If True, the stream specified by UserStream will be used to output the report. If multiple stream output is needed, like in graphic devices or in HTML, XHTML devices, provide an handler to the OnEndPage event. In this event, copy the contents of UserStream into another stream and then clear UserStream.
UserStream	**Type**: TStream Use UserStream to obtain the report in a stream instead of a file. UserStream will be used only if OutputToUserStream = True. If multiple stream output is needed, like in graphic devices or in HTML, XHTML devices, provide an handler to the OnEndPage event. In this event, copy the contents of UserStream into another stream and then clear UserStream.

Properties - Run-time and Design-time	
Active	**Type**: Boolean **Default**: True Specifies whether the Device is active. Specifies whether the Device is active. Devices are registered only if Active is True. Set Active to True inorder to export using the device. Values written through one automatically reflect in others.
BackgroundColor	**Type**: TColor **Default**: clWhite Indicates the color to be used as background color of all pages of report. Use BackgroundColor to specify the color to be used as background color of all pages of report.
BackgroundImage	**Type**: TPicture **Default**: nil BackgroundImage stores the Image that is to be used as background image of all pages of report. Use BackgroundImage to specify the Image to be used as background image of all pages of report. The background image will be displayed depending on the value of BackgroundImageDisplayType.
BackgroundImageDisplayType	**Type**: TPsRPBackgroundDisplay **Default**: dtCenter TPsRPBackgroundDisplay = (dtTile, dtTopLeft, dtTopCenter, dtTopRight, dtCenterLeft, dtCenter, dtCenterRight, dtBottomLeft, dtBottomCenter, dtBottomRight); BackgroundImageDisplayType specifies the way background image has to be displayed in the report. Use BackgroundImageDisplayType to specify the way background image has to be displayed in the report. BackgroundImageDisplayType will only be used when BackgroundImage has been set.

EMailAfterGenerate	**Type**: Boolean **Default**: True Specifies whether the report that is generated should be E-Mailed. Indicate whether the report generated should be E-Mailed. The OnEmail event is triggered just before mailing. Provide a handler to this event to set the required properties needed to send E-Mail. In order to send E-Mail, either Indy or FastNet components are used. **Note:** Enable the **EMailWithFastNet** or **EMailWithIndy** option in "PsRBDefines.inc" to get built-in support for FastNet or Indy Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
OpenAfterGenerate	**Type**: Boolean **Default**: True Specifies whether the report should be opened after it is generated. Indicate whether the report should be opened after it is generated.
Pages	**Type**: string **Default**: '' This property stores the pages that are to be exported. Pages property contains page numbers and/or page ranges separated by commas. For example, 1,3,5-12. If Pages is blank then all pages are exported.
RestrictToPages	**Type**: Boolean **Default**: True Specifies whether all the pages or the pages specified by the Pages property is to be exported. Specifies whether all the pages or only the pages specified by the Pages property is to be exported. If True, only the pages specified by the Pages property is exported. If Pages is blank, then the whole report is exported.

ShowProgress	**Type**: Boolean **Default**: True Specifies whether progress dialog should be displayed showing the current page that is being exported. Indicate whether progress dialog should be displayed showing the current page that is being exported. If True, then a progress dialog is displayed showing the current page that is being exported. If SilentMode is selected, then default value is False.
ShowSetupDialog	**Type**: Boolean **Default**: True Specifies whether Setup dialog should be displayed before exporting the report. Indicate whether Setup dialog should be displayed before exporting the report. If True, Setup dialog is displayed allowing the end user to set properties before exporting. If SilentMode is selected, then default value is False.
Version	**Type**: string Version stores the version details of the PsRBExportDevice components.
Events	

OnCancelJob	**Type**: TPsRBNotifyEvent
	
	TPsRBNotifyEvent = **procedure**(Sender: TRBExportDevice) **of object**;
	
	Triggers when an export operation is cancelled.
	
	Example:
	
	Delphi:
	
	`procedure TForm1.PsRBExportCancelJob (Sender: TPsRBExportDevice);`
	`begin`
	`// StatusBar1 should be there on TForm1.`
	`StatusBar1.SimpleText :=`
	`'Job cancelled';`
	`end;`

OnEMail

Type: TPsRBEMailEvent

TPsRBEMailEvent = **procedure**(Sender: TRRBExportDevice; EMailInfo: TPsRBEMailInfo; **var** Continue: Boolean) **of object**;

Triggers just before sending E-Mail. Provide a handler for this event to set E-Mail information which will then be used to E-Mail report files. E-Mail is sent using either FastNet or Indy components. Set Continue to True to trigger the event continuously so that multiple E-Mails can be sent. Set **EMailInfo.CancelEMail** to True inorder to cancel EMail operation after OnEMail event is triggered. Set **EMailInfo.DeleteFilesAfterEMail** to True to delete all disk files that were created during export process, after EMail operation is completed.

Note: Enable the **EMailWithFastNet** or **EMailWithIndy** option in "PsRBDefines.inc" to get built-in support for FastNet or Indy Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.

Example:

Delphi:

```
var
AHost, AUserID, APassword: string;
ASubject, AReplyTo: string;
AFromAddress, AFromName: string;
ABody, AToAddressList: TStringList;
ACCList, ABCCList: TStringList;

procedure TForm1.PsRBExportPDFEMail
(Sender: TRBExportDevice; EMailInfo:
TPsRBEMailInfo; var Continue: Boolean);
begin
with EMailInfo do
begin
Host := AHost;
UserID := AUserID;
Password := APassword;
Body.Assign(ABody);
Date := Now;
FromAddress := AFromAddress;
FromName := AFromName;
ReplyTo := AReplyTo;
Subject := ASubject;
RecipientList.Assign(AToAddress);
CCList. Assign(ACCList);
BCCList. Assign (ABCCList);
end;
end;
```

OnEMailConnect	**Type:** TNotifyEvent
	
	TNotifyEvent = **procedure**(Sender: TObject) **of object**;
	
	Note: This event is available only when **EMailWithFastNet** is enabled. Enable the **EMailWithFastNet** option in "PsRBDefines.inc" to get built-in support for FastNet Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
	
	Triggers when a connection is established.
	
	Example:
	
	Delphi:
	
	`procedure TForm1.PsRBExportEMailConnect (Sender: TObject);`
	`begin`
	`ShowMessage('Connected');`
	`end;`

OnEMailConnected	**Type:** TNotifyEvent
	
	TNotifyEvent = **procedure**(Sender: TObject) **of object**;
	
	Note: This event is available only when **EMailWithIndy** is enabled. Enable the **EMailWithIndy** option in "PsRBDefines.inc" to get built-in support for Indy Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
	
	Triggers when a connection is established.
	
	Example:
	
	Delphi:
	
	`procedure TForm1. PsRBExportEMailConnected(Sender: TObject);`
	`begin`
	`ShowMessage('Connected');`
	`end;`

OnEMailConnectionFalied	**Type:** TNotifyEvent
	
	TNotifyEvent = **procedure**(Sender: TObject) **of object**;
	
	Note: This event is available only when **EMailWithFastNet** is enabled. Enable the **EMailWithFastNet** option in "PsRBDefines.inc" to get built-in support for FastNet Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
	
	Triggers when an attempt to connect to server fails.
	
	Example:
	
	Delphi:
	
	`procedure TForm1.`
	`PsRBExportEMailConnectionFailed (Sender:`
	`TObject);`
	`begin`
	`ShowMessage('Connection Failed');`
	`end;`

OnEMailDisConnect	**Type:** TNotifyEvent
	
	TNotifyEvent = **procedure**(Sender: TObject) **of object**;
	
	Note: This event is available only when **EMailWithFastNet** is enabled. Enable the **EMailWithFastNet** option in "PsRBDefines.inc" to get built-in support for FastNet Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
	
	Triggers when a connection to a server is disconnected.
	
	Example:
	
	Delphi:
	
	`procedure TForm1.`
	`PsRBExportEMailDisConnect (Sender:`
	`TObject);`
	`begin`
	`ShowMessage('DisConnected');`
	`end;`

OnEMailDisConnected	**Type:** TNotifyEvent
	
	TNotifyEvent = **procedure**(Sender: TObject) **of object**;
	
	Note: This event is available only when **EMailWithIndy** is enabled. Enable the **EMailWithIndy** option in "PsRBDefines.inc" to get built-in support for Indy Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
	
	Triggers when a connection to a server is disconnected.
	
	Example:
	
	Delphi:
	
	`procedure TForm1.PsRBExportEMailDisConnected (Sender: TObject);`
	`begin`
	`ShowMessage('DisConnected');`
	`end;`

OnEMailError	**Type:** TPsEMailErrorEvent;
	
	TPsEMailErrorEvent = **procedure**(Sender: TPsRBExportDevice; const ErrMsg: string) **of object**;
	
	Note: This event is available only when **EMailWithIndy** is enabled.**:** Enable the **EMailWithIndy** option in "PsRBDefines.inc" to get built-in support for Indy Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
	
	Triggers when error is encountered while sending E-Mail.
	
	Example:
	
	Delphi:
	
	`procedure TForm1.PsRBExportEMailError (Sender: TPsRBExportDevice; const ErrMsg: string);`
	`begin`
	`ShowMessage('Error Occurred: ' + ErrMsg);`
	`end;`

OnEMailFailure	**Type:** TNotifyEvent;
	
	TNotifyEvent = **procedure**(Sender: TObject) **of object**;
	
	Note: This event is available only when **EMailWithFastNet** is enabled. Enable the **EMailWithFastNet** option in "PsRBDefines.inc" to get built-in support for FastNet Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
	
	Triggers when an attempt to send E-Mail fails.
	
	Example:
	
	Delphi:
	
	`procedure TForm1.PsRBExportEMailFailure (Sender: TObject);`
	`begin`
	`ShowMessage('E-Mail Failed');`
	`end;`

OnEMailInvalidHost	**Type**: THandlerEvent
	
	THandlerEvent = **procedure**(**var** Handled: Boolean) **of object**;
	
	Note: This event is available only when **EMailWithFastNet** is enabled. Enable the **EMailWithFastNet** option in "PsRBDefines.inc" to get built-in support for FastNet Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
	
	Triggers when Host specified is invalid.
	
	Example:
	
	Delphi:
	
	`procedure TForm1.`
	`PsRBExportPDFInvalidHost (var Handled: Boolean);`
	`begin`
	`Handled := False;`
	`ShowMessage('Host Invalid');`
	`end;`

OnEMailSendStart	**Type:** TNotifyEvent;
	
	TNotifyEvent = **procedure**(Sender: TObject) **of object**;
	
	Note: This event is available only when **EMailWithFastNet** is enabled. Enable the **EMailWithFastNet** option in "PsRBDefines.inc" to get built-in support for FastNet Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
	
	Triggers when mailing is started.
	
	Example:
	
	Delphi:
	
	`procedure TForm1.PsRBExportEMailSendStart(Sender: TObject);`
	`begin`
	`ShowMessage('Sending Mail');`
	`end;`

OnEMailStatus	**Type:** TIdStatusEvent; (Indy)
	Type: TOnStatus; (FastNet)
	
	EMailWithFastNet:
	TOnStatus = **procedure**(Sender: TComponent; Status: **string**) **of object**;
	
	EMailWithIndy:
	TIdStatusEvent = **procedure**(axSender: TObject; **const** axStatus: TIdStatus; **const** asStatusText: **string**) **of object**;
	
	Note: Enable the **EMailWithFastNet** or **EMailWithIndy** option in "PsRBDefines.inc" to get built-in support for FastNet or Indy Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
	
	Triggers when there is any status change.
	
	Example:
	
	Delphi:
	
	EMailWithFastNet:
	
	`procedure TForm1.PsRBExportEMailStatus(Sender: TComponent; Status: string);`
	`begin`
	`ShowMessage('Status = ' + Status);`
	`end;`

	EMailWithIndy:
	
	procedure TForm1.
	PsRBExportEMailStatus(axSender: TObject;
	const axStatus: TIdStatus; **const**
	asStatusText: **string**);
	begin
	ShowMessage(
	'Status = ' + asStatusText);
	end;

OnEMailSuccess	**Type:** TNotifyEvent;
	
	TNotifyEvent = **procedure**(Sender: TObject)
	of object;
	
	Note: This event is available with both **EMailWithFastNet** and **EMailWithIndy** options. Enable the **EMailWithFastNet** or **EMailWithIndy** option in "PsRBDefines.inc" to get built-in support for FastNet or Indy Components. "PsRBDefines.inc" is located in the directory <PsRBExportDevices>\Source.
	
	Triggers when E-Mail is successfully sent.
	
	Example:
	
	Delphi:
	
	procedure TForm1.PsRBExportEMailSuccess
	(Sender: TObject);
	begin
	ShowMessage('E-Mail Successful');
	end;

<table>
<tr><td>OnEncodeGraphic</td><td>Type: TPsRBEncodeGraphicEvent;

TPsRBEncodeGraphicEvent = procedure(Sender: TPsRBExportDevice; ADrawCommand: TppDrawCommand; var UniqueImage: Boolean; var AltText: string) of object;

This event is only available in HTML, XHTML and PDF Device. Triggers just before an Image is encoded. Provide a handler for this event to reuse Images. Assign a unique number to ADrawCommand.Tag property and say UniqueImage = False. All the Images with same tag will be considered as the same Image. A Image with a Tag value of 0 will not be reused.

Example:

<u>Delphi:</u>

<pre>var
AImageID: Integer;

procedure TForm1.
PsRBExportHTMLEncodeGraphic(Sender:
TPsRBExportDevice; ADrawCommand:
TppDrawCommand; var UniqueImage:
Boolean; var AltText: string);
begin
if ADrawCommand.Tag = AImageID then
UniqueImage := False;
end;</pre></td></tr>
<tr><td>OnEndJob</td><td>Type: TPsRBNotifyEvent;

TPsRBNotifyEvent = procedure(Sender: TPsRBExportDevice) of object;

Triggers at the end of the document. Provide a handler for this event to access the device object after the document has been exported.

Example:

<u>Delphi:</u>

<pre>procedure TForm1.PsRBExportEndJob
(Sender: TPsRBExportDevice);
begin
// StatusBar1 should be there on TForm1.
StatusBar1.SimpleText :=
'End of Job';
end;</pre></td></tr>
</table>

<table>
<tr><td>OnEndPage</td><td>Type: TPsRBNotifyEvent;

TPsRBNotifyEvent = procedure(Sender: TPsRBExportDevice) of object;

Triggers at the end of each page. Provide a handler for this event to access the device object at the end of each page.

Example:

<u>Delphi:</u>

<pre>var
      PageNo: Integer;

procedure TForm1.PsRBExportEndPage
(Sender: TPsRBExportDevice);
begin
// StatusBar1 should be there on TForm1.
StatusBar1.SimpleText :=
'End of Page' + IntToStr(PageNo);
end;</pre></td></tr>
<tr><td>OnMakeReportFileName</td><td>Type: TMakeReportFileNameEvent

TMakeReportFileNameEvent = procedure(Sender: TPsRBExportDevice; var FileName: string; PageNo: Integer) of object;

Triggers when a report filename is created. Provide a handler for this event to change the report file name. FileName specifies the name of the report file. PageNo specifies the current page that is begin exported.

Example:

<u>Delphi:</u>

<pre>procedure Form1.
PsRBExportHTMLMakeReportFileName
(Sender: TPsRBExportDevice; var
FileName: String; PageNo: Integer);
begin
FileName := ExtractFilePath(FileName) +
'TestReport' + IntToStr(PageNo) +
'.htm';
end;</pre></td></tr>
</table>

OnMakeImageFileName	**Type**: TMakeImageFileNameEvent
	
	TMakeImageFileNameEvent = **procedure**(Sender: TPsRBExportDevice; **var** FileName, AltText: **string**; PageNo: Integer) **of object**;
	
	This event is only available in HTML and XHTML Device. Triggers when a Image filename is made. Provide a handler for this event to change the image file name. FileName specifies the name of the image file. PageNo specifies the current page that is begin exported.
	
	Example:
	Delphi:
	
	`var`
	`ImageCount: Integer;`
	`procedure TForm1.`
	`PsRBExportHTMLMakeImageFileName (Sender: TPsRBExportDevice; var FileName, AltText: String; PageNo: Integer);`
	`begin`
	`Inc(ImgCount);`
	`FileName := ExtractFilePath(FileName) + 'TestImage' + IntToStr(ImgCount) + '.jpeg';`
	`AltText := FileName;`
	`end;`

TPsRBExportHTML = class(TPsRBExportComponent)	
Properties - Run-time only	
ImageFileCount	**Type**: Integer Query ImageFileCount to know the total number of image files that were created during the last export-to-HTML operation. The value is undefined after any other export.
ImageFileNames	**Type**: TStringList ImageFileNames is a TStringList object that stores the names of all image files that were created during the last export-to-HTML operation. Read ImageFileNames if you need the file names of all images that were created.
Properties - Run-time and Design-time	
Author	**Type**: string This property stores the name of the author of the report. Set the value for the Author Meta content, that will be encoded into the document. The value you usually specify is your name, the author of the report.
Creator	**Type**: string **Default**: 'ReportBuilder' Creator stores the name of the tool used to create the report. Specifies the tool that was used for creating the report. Rightly, it should always be 'ReportBuilder'.
CSSFileName	**Type**: string Stores the name of the CSS file. Indicate the name of the CSS file. The default name of CSS file is name or report file. CSS file is created only when OptimizeForIE = True and OutputStylesInCSSFile = True. Having an external CSS file will help in reducing the size of HTML files.
DefaultFont	**Type**: TFont **Default**: 'Arial' Stores the default Font that will be used to encode in HEAD or CSS File. Indicates the default Font used to encode in HEAD or CSS File. The Default value is used only when OptimizeForIE = True.

DocumentPreferences	**Type**: TPsHTMLDocumentPreferences **TPsXHTMLDocumentPreferences** = **class** DisableRightClick: Boolean; DisableCopying: Boolean; AutoMaximizeWindow: Boolean; **end;** Specifies the various options relating to the XHTML document.
DisableRightClick	**Type:** Boolean **Default:** False **Description**: Set DisableRightClick to true to disable the right click popup menu.
DisableCopying	**Type:** Boolean **Default:** False **Description**: Set DisableCopying to true to disallow copying of the contents of the XHTML file
AutoMaximizeWindow	**Type:** Boolean **Default**: False **Description**: AutoMaximizeWindow specifies whether the XHTML document should be automatically maximized when opened. If the value is True then the XHTML document is automatically maximized when opened.
ExportImageFormat	**Type**: TPsRPImageFormat **Default**: ifJPG TPsRBExportImageFormat = (ifGIF, ifJPG, ifBMP) Specifies the Linked image file format for image controls used in the report. Use ExportImageFormat to specify the linked image file format for image controls used in the report.

GenerateTOC	**Type**: Boolean
	Default: False
	
	Specifies whether Table of Contents should be encoded in the HTML document.
	
	Specifies whether Table of Contents should be encoded in the HTML document. Set GenerateTOC to True to generate a table of contents and encode it in HTML document. Use the public method AddTOC to add a specific entry into the Table of Contents. If no TOC is specified then by default an entry for each page is added to the TOC list.
	
	`procedure AddTOC(ALinkText: string; APageNo: Integer);`
	
	ALinkText specifies the text representing the particular entry into the TOC list. APageNo specifies the page to be opened when the link is clicked.
	
	Example (Delphi):
	
	`var`
	`APageNo: Integer;`
	`procedure TForm1.PsRBExportBeginPage(Sender: TPsRBExportDevice);`
	`begin`
	`Inc(APageNo);`
	`PsRBPDFSettings.AddTOC('Page' + IntToStr(APageNo), APageNo);`
	`end;`

ImageFolder	**Type**: string Stores the name of the folder where Image files should be created in. Indicate the name of the folder, image files should be created in. The default folder for image files is the folder in which the main HTML file was saved. Setting ImageFolder is a convenient way of separating the HTML from the concomitant linked files. Specifying ImageFolder will not cause any loss of HTML content, when the HTML and the image folder are copied to another location, as the images are encoded with relative path information (relative to the location of the HTML). If the path name specified in ImageFolder does not exist an attempt is made to create the folder(s). If the string specified is not a valid path name images are created in the same folder as the HTML.

ImageDPI	**Type**: Integer **Default**: -1 Indicates DPI value to be used while encoding images. Indicates DPI value to be used while encoding images. The quality of Image can be enhanced by giving a greater value to ImageDPI**.** But increase in the value of ImageDPI will lead to increase in file size. ImageDPI is not considered when ImageDPI = -1, i.e the images are exported as is without any modifications in quality.
ImagePixelFormat	**Type**: TPixelFormat **Default**: pf24bit **TPixelFormat** = (pfDevice, pf1bit, pf4bit, pf8bit, pf15bit, pf16bit, pf24bit, pf32bit, pfCustom); Specifies the pixel format of the image. Indicate the pixel format of the image. Setting **ImagePixelFormat** will change the internal image and color depth of the image.
IncludeHyperlinks	**Type**: Boolean **Default**: True Specifies whether all Internet addresses should be converted to hyperlinks. Indicate whether all Internet addresses should be converted to hyperlinks. Set IncludeHyperlinks to True to convert all Internet addresses in the report to hyperlinks.
IncludeImages	**Type**: Boolean **Default**: True Specifies whether Images on the report should be included in the exported file. Indicate whether images on the report should be included in the exported file. Setting IncludeImages to True includes images; setting False excludes them.
IncludeLines	**Type**: Boolean **Default**: True Specifies whether lines on the report should be included in the exported file. Indicate whether lines on the report should be included in the exported file. Setting IncludeLines to True includes lines; setting to False excludes them.

IncludeRichText	**Type**: Boolean **Default**: True Specifies whether contents of the RichText objects on the report should be included in the exported file. Indicate whether RichText contents on the report should be included in the exported file. Setting IncludeRichText to True includes RichText contents; setting False excludes them. In PDF, HTML, XHTML, Excel devices, RichText objects are exported either as image or as plain text based on the value of RichTextEncodingType property.
IncludetShapes	**Type**: Boolean **Default**: True Specifies whether shapes on the report should be included in the exported file. Indicate whether shapes on the report should be included in the exported file. Setting IncludeShapes to True includes shapes; setting to False excludes them.
JavaScriptFileName	**Type**: string Specifies the name of the JavaScript file. Indicate the name of the JavaScript file. The default name of JavaScript file is name of the report file. Having an external JavaScript file will help in reducing the size of HTML files.
JPEGQuality	**Type**: TJPEGQualityRange **Default**: 100 TJPEGQualityRange = 1..100; Specifies the Quality of the JPEG Image Used when ExportImageFormat = ifJPG. Specifies the quality of the JPEG image
Keywords	**Type**: string **Default**: '' Keywords stores the list of keywords used in the report. Specify a list of keywords for the document that should be encoded as the Keywords Meta content.

LinkTextFirst **LinkTextNext** **LinkTextPrev** **LinkTextLast**	**Type**: string **Default**: 'First' **Type**: string **Default**: 'Next' **Type**: string **Default**: 'Previous' **Type**: string **Default**: 'Last' Indicate the text that should be used as the caption for the page navigator links that appear at the bottom of the produced HTML. The LinkText values are used only when ShowNavigator = True, UseTextLinks = True and SeparateFilePerPage = True.
LinkBackColor **LinkHoverForeColor** **LinkHoverBackColor**	**Type**: TColor **Default**: clWhite **Type**: TColor **Default**: clWhite **Type**: TColor **Default**: clBlue Indicates the color of the navigator when the mouse pointer hovers on it. The LinkColor values are used only when ShowNavigator = True and SeparateFilePerPage = True.
LinkImgSRCFirst **LinkImgSRCNext** **LinkImgSRCPrev** **LinkImgSRCLast**	**Type**: string **Type**: string **Type**: string **Type**: string Indicates the Filenames of the images to be used as navigator images. The LinkImgSRC values are used only when ShowNavigator = True, UseTextLinks = False and SeparateFilePerPage = True.
LinkFont	**Type**: TFont **Default**: Name = 'Wingdings', Color = clBlue Stores the Font of the LinkText. Indicates the Font of the LinkText used to represent the navigator. The LinkFont value is used only when ShowNavigator = True, UseTextLinks = True and SeparateFilePerPage = True.

NavigatorOrientation	**Type**: TPsRBNavigatorOrientation **Default:** noHorizontal TPsRBNavigatorOrientation = (noHorizontal, noVertical); Specifies the orientation of the navigator. Specifies the orientation of the navigator. if NavigatorOrientation = noHorizontal then, a Horizontal navigator is encoded, else a Vertical navigator is encoded. NavigatorOrientation will be considered only when ShowNavigator = True and SeparateFilePerPage = True.
NavigatorPosition	**Type**: TPsRBNavigatorPosition **Default:** npRightBottom TPsRBNavigatorPosition = (npTopLeft, npTopCenter, npTopRight, npCenterLeft, npCenter, npCenterRight, npBottomLeft, npBottomCenter, npBottomRight); Specifies the position of the navigator. Specifies the position of the navigator. NavigatorPosition will be considered only when ShowNavigator = True and SeparateFilePerPage = True.
NavigatorType	**Type**: TPsRBNavigatorType **Default**: ntFixedToScreen TPsRBNavigatorType = (ntFixedToScreen, ntFixedToPage); Specifies the type of the navigator. Specifies the type of the navigator. if NavigatorType = ntFixedToScreen then the navigator will be fixed to screen, else the navigator will be fixed to page. NavigatorType will be considered only when ShowNavigator = True and SeparateFilePerPage = True.
OptimizeForIE	**Type**: Boolean **Default**: True Specifies whether HTML report should be optimized for Internet Explorer. Indicates whether the HTML report should be optimized for Internet Explorer. Setting OptimizeForIE to True includes CSS styles into HTML file. CSS styles are encoded in separate CSS file if OutputStylesToCSSFile = True, if OutputStylesToCSSFile = False CSS styles are embedded in HEAD.

OutputScriptsToJavaScriptFile	**Type**: Boolean **Default**: True Default: True Specifies whether java scripts are to be stored in external file or to be embedded in HEAD. Indicates whether java scripts are to be stored in external file or to be embedded in HEAD. Setting OutputScriptsToJavaScriptFile = True will encode java scripts in external file. The name of the external file can be specified by JavaScriptFileName. if OutputScriptsToJavaScriptFile = False, java scripts are embedded in HEAD.
OutputStylesToCSSFile	**Type**: Boolean **Default**: True Specifies whether CSS styles are to be stored in external file or to be embedded in HTML document. Indicates whether CSS styles are to be stored in external file or to be embedded in HEAD of HTML document. Setting OutputStylesToCSSFile = True will encode CSS styles in external file. The name of the external file can be specified by CSSFileName. If OutputStylesToCSSFile = False CSS styles are embedded in HEAD.
PageEndLines	**Type**: Boolean **Default**: True Specifies whether a line should be drawn at the end of each page. Indicates whether a line should be drawn at the end of each page. Setting PageEndLines to True draws a line after each page. PageEndLines is considered only when SeparateFilePerPage = False.

RichTextEncodingType	**Type**: TPsRichTextEncodingType **Default**: rtImage **TPsRichTextEncodingType =** (rtImage, rtPlainText); Indicates the way in which RichText contents should be encoded in HTML document. Specify the way in which RichText contents should be encoded in exported HTML document. Set RichTextEncodingType to rtImage to encode RichText contents as exact graphical representations. Set RichTextEncodingType to rtPlainText to encode RichText contents as unformatted text. The font used when encoding RichText as plain text is the one set in the Font property of the RichText object. When RichTextEncodingType is set to rtPlainText make sure you set the SetDefaultFont property of the RichText object to true and set the required font in the Font property.
SeparateFilePerPage	**Type**: Boolean **Default**: True Specifies whether each page of the report should be stored in a separate file. Choose how the exported HTML content should be stored. Setting SeparateFilePerPage to False results in a single HTML file which stores the entire report, optionally marked by horizontal lines at the end of each page. Setting to True produces separate HTML files for each page of the report with links to navigate between pages, optionally.
ShowNavigator	**Type**: Boolean **Default**: True Specifies whether a Navigator should be included at the end of each page. Set to True to show a collection of hyperlinks that enable easy navigation between pages of the exported HTML. The links function much like the navigator buttons in the standard preview of ReportBuilder: one link to jump to the first page; one to the last page; one to the next page and another to get back to the previous page. ShowNavigator is only effective when SeparateFilePerPage = True.

Subject	**Type**: string Stores the subject of the report. Specify a subject of the report that should be encoded as the Subjects Meta content.
Title	**Type**: string Stores the Title of the report. Specify a Title of the report that should be encoded as the Title Meta content.
TOCFileName	**Type**: string Stores the Table of Contents filename. Specify the Table of Contents filename. Table of Contents is generated only if GenerateTOC is True. If TOCFilename is blank then the default filename is <Report FileName>_Index.htm
UseTextLinks	**Type**: Boolean **Default**: False Specifies whether the navigator is constructed using Textual hyperlinks. Indicate the type of the navigator links that should be encoded for a multi-file HTML output. Setting UseTextLinks to True uses the text provided in LinkTextFirst, LinkTextLast, LinkTextNext and LinkTextPrev as the hyperlink's caption. Setting to False uses Images specified by LinkImgSRCFirst, LinkImgSRCNext, LinkImgSRCPrev and LinkImgSRCLast. The property is effective only when ShowNavigator = True and SeparateFilePerPage = True.

TPsRBExportXHTML = class(TPsRBExportComponent)	
Properties - Run-time only	
ImageFileCount	Same as TPsRBExportHTML
ImageFileNames	
Properties - Run-time and Design-time	
Author	Same as TPsRBExportHTML
Creator	
CSSFileName	
DefaultFont	
DocumentPreferences	
GenerateTOC	
ExportImageFormat	
TOCFileName	
ImageFolder	
ImageDPI	
ImagePixelFormat	
IncludeHyperlinks	
IncludeImages	
IncludeLines	
IncludeRichText	
IncludetShapes	
JavaScriptFileName	
JPEGQuality	
Keywords	
LinkTextFirst **LinkTextNext** **LinkTextPrev** **LinkTextLast**	
LinkBackColor **LinkHoverForeColor** **LinkHoverBackColor**	
LinkImgSRCFirst **LinkImgSRCNext** **LinkImgSRCPrev** **LinkImgSRCLast**	
LinkFont	
NavigatorOrientation	
NavigatorPosition	
NavigatorType	
OutputScriptsToJavaScriptFile	
OutputStylesToCSSFile	
PageEndLines	
RichTextEncodingType	
SeparateFilePerPage	
ShowNavigator	
Subject	
Title	
UseTextLinks	

TPsRBExportPDF = class(TPsRBExportComponent)	
Properties - Run-time and Design-time	
Author	Same as TPsRBExportHTML.
Creator	
GenerateTOC	
ImageDPI	
ImagePixelFormat	
IncludeHyperlinks	
IncludeImages	
IncludeLines	
IncludeRichText	
IncludeShapes	
JPEGQuality	
Keywords	
RichTextEncodingType	
Subject	
Title	
CompressionMethod	**Type**: TPDFCompressionMethod **Default**: cmMaxCompress TPDFCompressionMethod = (cmFastest, cmNormal, cmMaxCompress); Specifies the type of compression to be used to compress text and Image. Indicate the type of compression to be used to compress text and Image. CompressionMethod will be used only if UseCompression = True. Compressing will help in reducing file size. Images are compressed only when ExportImageFormat = ifBMP. **Note:** In order to use compression, enable the **ZLib** option in "PsRBDefines.inc" located in <PsRBExportDevices>\Source directory.
EmbedUsedFonts	**Type**: Boolean **Default**: False Specifies whether all the used fonts should be embedded in the PDF document. Indicate whether all the used fonts should be embedded in the PDF document. By default fonts are embedded only when the font charset is not ANSI_CHARSET. Set EmbedUsedFonts to True to embed all the fonts used in the report. Fonts are embedded only if embedding is permitted depending on the Embedding Flags of the font.

EncryptFile	**Type**: Boolean **Default**: False Specifies whether the PDF document should be encrypted. Indicate whether the PDF document should be encrypted. Set EncryptFile to true to encrypt the PDF document.
EncryptionLevel	**Type**: TPsRPImageFormat **Default**: el40Bit TPsPDFEncryptionLevel = (el40Bit, el128Bit); Specifies the level of encryption of the PDF document. Indicate the level of encryption of the PDF document. Set EncryptionLevel to el40Bit for a 40 bit encryption of the PDF document, i.e a 40 bit key is used to encrypt the PDF document. Set EncryptionLevel to el128Bit for a 128 bit encryption of the PDF document, i.e a 128 bit key is used to encrypt the PDF document. EncryptionLevel is considered only if EncryptFile is True.
ExportImageFormat	**Type**: TPsRBExportImageFormat **Default**: ifBMP TPsRBExportImageFormat = (ifGIF, ifJPG, ifBMP) Note: Images within the exported PDF document can be of two formats: JPEG or BMP. These are the only formats currently supported. Specifies the Linked image file format for image controls used in the report. Use ExportImageFormat to specify the linked image file format for image controls used in the report.
Encoding	**Type**: TPDFFontEncoding **Default**: feWinAnsiEncoding TPDFFontEncoding = (feWinAnsiEncoding, feMacRomanEncoding, feMacExpertEncoding, feStandardEncoding, fePDFDocEncoding); Specifies the type of font encoding to be used to encode font. Indicate the type of font encoding to be used to encode font.

OwnerPassword	**Type**: string Specifies the owner password of the encrypted PDF document. Indicate the owner password of the encrypted PDF document. Opening the document with the correct owner password (assuming it is not same as the user password) allows full (owner) access to the document. This unlimited access includes the ability to change the document's passwords and access permissions.
Permissions	**Type**: TPsPDFPermissions TPsPDFPermissions = **class** EnablePrinting: Boolean **default**: True; EnableCopying: Boolean **default**: True; EnableModifying: Boolean **default**: True; **end**; Specifies the user permissions that should be encoded in the PDF document. Indicate the user permissions that should be encoded in the PDF document. These permissions are considered by the viewer applications if the document is opened with the User password. If EnablePrinting is False then printing is not enabled. If EnableCopying is False then page contents cannot be selected and copied. If EnableModifying is False then the PDF document cannot be modified.
UseCompression	**Type**: Boolean **Default**: True Specifies whether compression should be used. Set to True to compress text and Image while encoding. Text and Image will be compressed by method specified by CompressionMethod. Compressing will help in reducing file size. Images are compressed only when ExportImageFormat = ifBMP. **Note:** In order to use compression, enable the **ZLib** option in "PsRBDefines.inc" located in <PsRBExportDevices>\Source directory.

UserPassword	**Type**: string Specifies the user password of the encrypted PDF document. Specifies the user password of the encrypted PDF document. Opening the document with the correct user password allows additional operations to be performed according to the user access permissions specified by the Permissions property.
ViewerPreferences	**Type:** TPsPDFViewerPreferences **Scope**: Published TPsPDFViewerPreferences = **class** PageLayout: TPsPDFPageLayout; PageMode: TPsPDFPageMode; HideToolBar: Boolean; HideMenubar: Boolean; HideWindowsUI: Boolean; FitWindow: Boolean; CenterWindow: Boolean; NonFullScreenPageMode: TPsPDFNonFullScreenPageMode; PresentationMode: TPsPDFPresentationMode; **end**; **Description**: This property specifies the various viewer preferences of Document.
PageLayout	**Type:** TPsPDFPageLayout **Valid Values**: plSinglePage, plOneColumn, plTwoColumnLeft, plTwoColumnRight **Default:** plSinglePage **Description**: PageLayout specifies the page layout to be used when the PDF document is opened. If the value is plSinglePage then one page is displayed at a time. If the value is plOneColumn then pages are displayed in one column. If the value is plTwoColumnLeft then pages are displayed in two columns with odd numbered pages on the left. If the value is plTwoColumnRight then pages are displayed in two columns with odd numbered pages on the right.
PageMode	**Type:** TPsPDFPageMode **Valid Values**: pmUseNone, pmUseOutlines, pmUseThumbs, pmFullScreen **Default:** pmUseNone **Description**: PageMode specifies how the document should be displayed when opened. If the value is pmUseNone then neither document outlines nor thumbnail images are visible. If the value is pmUseOutlines then document outline is visible. If the value is pmUseThumbs then thumbnail images are visible. If the value is pmFullScreen then document is opened in Full

	Screen mode with no menu bar, window controls, or any other window visible.
HideToolbar	**Type:** Boolean **Valid Values**: True, False **Default**: False **Description**: This is a flag specifying whether to hide the viewer application's toolbars when the document is active. If the value is True then the viewer applications toolbars are hidden.
HideMenubar	**Type:** Boolean **Valid Values**: True, False **Default**: False **Description**: This is a flag specifying whether to hide the viewer application's menu bar when the document is active. If the value is True then the viewer applications menu bar is hidden.
HideWindowUI	**Type:** Boolean **Valid Values**: True, False **Default**: False **Description**: This is a flag specifying whether to hide user interface elements in the document's window (such as scroll bars and navigation controls), leaving only the document's contents displayed. If the value is True then documents user interface elements are hidden.
FitWindow	**Type:** Boolean **Valid Values**: True, False **Default**: False **Description**: This is a flag specifying whether to resize the document's window to fit the size of the first displayed page. If the value is True then the first page displayed is resized to fit the document's window.
CenterWindow	**Type:** Boolean **Valid Values**: True, False **Default**: False **Description**: This is a flag specifying whether to position the document's window in the center of the screen. If the value is True then the document is positioned at the center of the screen when opened.
NonFullScreenPageMode	**Type:** TPsPDFNonFullScreenPageMode **Valid Values**: fsUseNone, fsUseOutlines, fsUseThumbs **Default**: pmUseNone **Description**: NonFullScreenPageMode specifies how to display the document on exiting full screen mode. This is meaningful only if the PageMode entry is pmfullScreen. If the value is fsUseNone then neither document outlines nor

	thumbnail images are visible. If the value is fsUseOutlines then document outline is visible. If the value is fsUseThumbs then thumbnail images are visible.
PresentationMode	**Type:** TPsPDFPresentationMode TPsPDFPresentationMode = **class** Duration: Integer; TransitionEffects: TPsPDFTransitionEffects; **end**; TPsPDFTransitionEffects = (teSplitHorizontalInward, teSplitHorizontalOutward, teSplitVerticalInward, teSplitVerticaloutward, teBlindsHorizontal, teBlindsVertical, teBoxInward, teBoxOutward, teWipeLeftToRight, teWipeBottomToTop, teWipeRightToLeft, teWipeTopToBottom, teDissolve, teGlitterLeftToRight, teGlitterTopToBottom, teGlitterTopLeftToBottomRight, teNone); **Description**: PresentationMode specifies how the viewer application should display the document. Some viewer applications allow a document to be displayed in the form of a presentation or "slide show," advancing from one page to the next either automatically or under user control. Duration specifies the duration of the transition effect in seconds. TransitionEffects specify how to display each page in presentation mode. The various TransitionEffects are as follows: **teSplitHorizontalInward**: Two lines sweep across the screen, revealing the new page. The lines are horizontal and move inward from the edges of the page. **teSplitHorizontalOutward**: Two lines sweep across the screen, revealing the new page. The lines are horizontal and move outward from the center of the page. **teSplitVerticalInward**: Two lines sweep across the screen, revealing the new page. The lines are vertical and move inward from the edges of the page. **teSplitVerticaloutward**: Two lines sweep across the screen, revealing the new page. The lines are vertical and move outward from the center of the page. **teBlindsHorizontal**: Multiple lines, evenly spaced across the screen, synchronously sweep in the same direction to reveal the new page. The lines are horizontal and move downward. **teBlindsVertical**: Multiple lines, evenly spaced across the screen, synchronously sweep in the same direction to reveal the new page. The lines are vertical and move to the right. **teBoxInward**: A rectangular box sweeps

	inward from the edges of the page revealing the new page. **teBoxOutward**: A rectangular box sweeps outward from the center of the page revealing the new page. **teWipeLeftToRight**: A single line sweeps across the screen from Left to Right edge of the page **teWipeBottomToTop**: A single line sweeps across the screen from Bottom to the Top edge of the page **teWipeRightToLeft**: A single line sweeps across the screen from Right to the Left edge of the page **teWipeTopToBottom**: A single line sweeps across the screen from Top to Bottom edge of the page **teDissolve**: The old page "dissolves" gradually to reveal the new one. **teGlitterLeftToRight**: Similar to Dissolve, except that the effect sweeps across the page in a wide band moving from Left side of the screen to the Right side. **teGlitterTopToBottom**: Similar to Dissolve, except that the effect sweeps across the page in a wide band moving from Top side of the screen to the Bottom side. **teGlitterTopLeftToBottomRight**: Similar to Dissolve, except that the effect sweeps across the page in a wide band moving from TopLeft corner of the screen to the BottomRight corner. **teNone**: The new page simply replaces the old one with no special transition effect

TPsRBExportRTF = class(TPsRBExportComponent)	
Properties - Run-time and Design-time	
Author	Same as TPsRBExportHTML.
Creator	
ImageDPI	
ImagePixelFormat	
IncludeImages	
IncludeLines	
IncludeRichText	
IncludeShapes	
JPEGQuality	
Keywords	
Subject	
Title	
OptimizationMode	**Type**: TPsRBRTFOptimizationMode **Default:** omPositionalAccuracy TPsRBRTFOptimizationMode = (omPositionalAccuracy, omEaseOfEditing); Specifies the type of encoding. Indicate the type of encoding of the RTF Document. If OptimizationMode = omPositionalAccuracy then absolute positioning of the text is done, else tabs are introduced to position text.
ExportImageFormat	**Type**: TPsRPImageFormat **Default**: ifJPG TPsRBExportImageFormat = (ifGIF, ifJPG, ifBMP) Note: Images within the exported RTF document can be of two formats: JPEG or BMP. These are the only formats currently supported. Specifies the Linked image file format for image controls used in the report. Use ExportImageFormat to specify the linked image file format for image controls used in the report.
GraphicDataInBinary	**Type**: Boolean **Default**: True Specifies whether graphic data is to be encoded in binary format. Set to True to encode graphic data in binary format. If False, graphic data is encoded in hexadecimal format.

TPsRBExportExcel = class(TPsRBExportComponent)	
Properties - Run-time and Design-time	
IncludeImages	Same as TPsRBExportHTML.
IncludeLines	
IncludeRichText	
IncludeShapes	
RichTextEncodingType	
AllTextInGeneralFormat	**Type**: Boolean **Default**: False Specifies whether all the text in the report should be converted to general format or the format of the text should be detected and encoded accordingly. ndicate whether all the text in the report should be converted to general format or the format of the text should be detected and encoded accordingly. Set AllTextInGeneralFormat to True encode all the text contents in the report in general format. General format text have no specific number format. Set AllTextInGeneralFormat to False to detect the format of the text and encode accordingly. In this case number and other formats are identified and encoded. This helps to include formulae in the exported excel document.
ColumnSpacing	**Type**: TPsTextSpacing **Default**: lsActual TPsTextSpacing = (lsActual, lsNoBlank, lsOneBlank, lsTwoBlank, lsThreeBlank, lsFourBlank, lsFiveBlank); Specifies the spacing to be used between Columns of text. Indicate the amount of spacing between Columns.
ExportImageFormat	**Type**: TPsExportImageFormat **Default**: ifWMF TPsExportImageFormat = (ifBMP, ifWMF); Specifies the Linked image file format for image controls used in the report. Use ExportImageFormat to specify the linked image file format for image controls used in the report.

LineSpacing	**Type**: TPsTextSpacing **Default**: lsNoBlank TPsTextSpacing = (lsActual, lsNoBlank, lsOneBlank, lsTwoBlank, lsThreeBlank, lsFourBlank, lsFiveBlank); Specifies the spacing to be used between lines. Indicate the amount of spacing between lines.
OneSheetPerPage	**Type**: Boolean **Default**: False Specifies whether each page in the report should be encoded into separate excel sheets or all the pages should be encoded in a single excel sheet. Indicate whether each page in the report should be encoded into separate excel sheets or all the pages should be encoded in a single excel sheet.
PositionalGridDimensions	**Type**: TPsPositionalGridDimensions TPsPositionalGridDimensions = **class** // 16934 mm thousandths = 64 pixels approx. Width: Integer default 16934; // 4234 mm thousandths = 16 pixels approx. Height: Integer default 4234; **end**; Specifies the positional grid dimensions used to calculate the Row and Column of elements of the report. Indicate the positional grid dimensions used to calculate the Row and Column of elements of the report. The PositionalGridDimensions are expressed in mm thousandths. To stretch (or space out) the export along the horizontal axis, set a lesser value to the Width. Conversely, to compress the export width-wise set a greater value to width. Similarly to stretch (or space out) the export along the vertical axis, set a lesser value to height. Conversely, to compress the export height-wise set a greater value to width. Setting a lesser value to the Width or Height means that the report is divided into more number of rows or columns. Similarly setting a greater value to the Width or Height means that the report is divided into less number of rows or columns.

TPsRBExportBMP = class(TPsRBExportComponent)	
Properties - Run-time and Design-time	
Monochrome	**Type**: Boolean **Default**: False Specifies whether images produced are color or monochrome images. Setting the property to True produces a monochrome image file. Setting to False produces color images of the report.
PixelFormat	**Type**: TPixelFormat **Default:** pf24bit TPixelFormat = (pfDevice, pf1bit, pf4bit, pf8bit, pf15bit, pf16bit, pf24bit, pf32bit, pfCustom); Specifies the pixel format of the Bitmap image. Indicate the pixel format of the Bitmap image. Setting PixelFormat will change the internal image and color depth of the bitmap.
ScaleX	**Type**: Double **Default**: 1 Scaling factor for the width of the exported page (X Axis).
ScaleY	**Type**: Double **Default**: 1 Scaling factor for the height of the exported page (Y Axis).

TPsRBExportJPEG = class(TPsRBExportComponent)	
Properties - Run-time and Design-time	
PixelFormat	Same as TPsRBExportBMP
ScaleX	
ScaleY	
Grayscale	**Type**: Boolean **Default**: False Determines whether the image output of a JPEG image is black and white, or color. When Grayscale is set to True, the color is separated from luminosity and the output contains 255 shades of gray instead of colors. Grayscale can also be set to True if speed of decompression is important.
ProgressiveEncoding	**Type**: Boolean **Default**: True Set ProgressiveEncoding to specify whether the JPEG image should be encoded in a manner such that it can be decompressed and displayed progressively. It is particularly useful, to set to True, for large files, so the user won't be shown a blank screen while waiting for the whole image to decompress.
Quality	**Type**: TJPEGQualityRange **Default**: 100 TJPEGQualityRange = 1..100; This property determines the quality of the saved JPEG images. As a side effect, quality of image is proportional to the size of the JPEG file. Setting a larger value improves image quality but also increases file size; a smaller value decreases image quality and also reduces file size. Alter it according to the need.

TPsRBExportGIF = class(TPsRBExportComponent)	
Properties - Run-time and Design-time	
Monochrome	Same as TPsRBExportBMP
PixelFormat	
ScaleX	
ScaleY	
TPsRBExportEMF = class(TPsRBExportComponent)	
ScaleX	Same as TPsRBExportBMP
ScaleY	
TPsRBExportWMF = class(TPsRBExportComponent)	
ScaleX	Same as TPsRBExportBMP
ScaleY	

Ordering and Delivery

Ordering

You can order **PsRBExportDevices** online over the Internet at http://www.rarefind.com/rbpro/buy.html.

PLEASE MAKE SURE you give a working E-Mail at the time of ordering.

Delivery

Delivery is **INSTANT** for credit card transactions. For Bank transfers and other modes of payment, delivery is upon receipt of full payment.

As soon as your order is passed you should receive an E-Mail giving you the download link to the registered version Setup program. Please follow the steps listed in the E-Mail to download the registered version.

If you are having problems downloading or installing PsRBExportDevices please visit our product support link at www.rarefind.com/rbpro/index.html

Contact

If you have any queries or feedback on the components please feel free to contact us:

Rarefind Engineering Innovations Pvt. Ltd.
Pragnaan Software Group
#4510, 5th Floor
High Point – 4
Palace Road
Bangalore - 560 001
INDIA
Web: http://www.rarefind.com/